

УДК004.774.6

АВТОМАТИЗАЦІЯ СЦЕНАРІЇВ ТЕСТУВАННЯ ВЕБ-ПОРТАЛУ

*Васильєв В.М., Осадчий В.В., Єремєєв В.С.
Мелітопольський державний педагогічний університет
імені Богдана Хмельницького, м. Мелітополь
e-mail: valeriyit108@gmail.com*

Постановка проблеми. В процесі проектування та розробки програмного забезпечення, в особливості при розробці web-порталу, виникають труднощі з виявленням та документуванням помилок. Особливо така проблема постає при розробці великих інформаційних систем, web-порталів та Інтернет магазинів. Головне завдання професійного розробника програмного забезпечення - створення в максимально короткий термін програмного продукту, яке б мало можливість повністю задовольняти замовника. Бібліотеки та скрипти повинні робити саме те, чого від них очікують. Найпростіший приклад - форма зворотного зв'язку, яка після натискання на кнопку за умови коректного заповнення полів відправляє пошту або записує дані в таблицю БД [2].

Тому при розробці Web-порталу обліку міграції птахів в Азово-Чорноморському регіоні України в плані тестування постали такі проблеми:

- необхідність перевіряти коректність роботи різних частин програмного продукту в процесі розробки;
- необхідність відслідковувати швидкість та стабільність роботи системи з метою подальшої оптимізації;
- необхідність перевіряти коректність роботи заповнення форм, для уникнення помилок з вхідними даними, включаючи вивід повідомлень про помилку або успішне збереження даних;
- можливість перевірити систему на безпечність та захищеність від SQL-ін'єкцій [3].

Мета роботи: зробити аналіз існуючих типів автоматизованих тестів, розробити їх структуру та сценарії

Виклад основного матеріалу. Для визначення проблемних ділянок web-порталу використовується тестування - діяльність, спрямована на виявлення невідповідностей між очікуваним і дійсним. Виявляючи некоректну поведінку на стадії розробки, розробник свідомо і плановірно зменшує ймовірність того, що з цим доведеться зіткнутися користувачеві [4]. Тестувати програми (зокрема php-скрипти) цілком можна і вручну. На прикладі форми зворотного зв'язку, зрозуміло, що протестувати її дуже просто. Запустити браузер, зайти на потрібний url, заповнити поля, натиснути на "Ок", потім зайти в базу і перевірити, чи з'явився там відповідний запис.

Автоматизовані тести можна розділити на окремі типи залежно від свого призначення: функціональні тести, навантажувальні тести, GUI тести, тестування безпеки, тестування продуктивності, Unit-тести, тестування коду, тестування баз даних [5].

Також, варто віддавати собі звіт в тому, що автоматизація тестування ПЗ- це більше, ніж просто виконання тестів різних видів [6]. Джеймс Бах в своїй статті «Agile Test Automation» наводить наступний список завдань, у вирішенні яких в якійсь мірі можна застосувати автоматизацію:

- генератор тестів (генератор даних і скриптів). Інструменти автоматизації можуть створювати спеціальні дані, наприклад, довільні електронні листи, або наповнювати базу даних, або генерувати комбінації параметрів і даних, які нам необхідні при тестуванні;
- виконання тестів (середина і тестові скрипти). Інструменти можуть управляти додатком що тестується самостійно, або симулювати роботу користувача з графічним інтерфейсом додатки, або обходячи графічний, використовувати альтернативний тестований інтерфейс [1];
- дослідження. Інструменти можуть показати те, що залишилося б невидимим для людей. Вони можуть статично аналізувати продукт, парсити файли з логами, або проводити моніторинг параметрів системи [7].

Структура додатку для проведення автоматизованого тестування web-додатку обліку міграції птахів складається із розділів:

- HomeCept – тестування запуску головної сторінки;
- SignupCept – тестування форми реєстрації;
- LoginCept – тестування форми авторизації;
- ContentviewCept – тестування сторінки з довідковою інформацією;
- BirdsguidindexCept – тестування сторінки з довідником о птахів;
- PlaceguidCept – тестування довідника з даними о площадках спостережень;
- WatchpointCept – тестування довідника з даними о пунктах спотосрежень;
- CreatepointviewCept – тестування форми вводу інформації з умовами спостережень за птахами на пунктах спостереження;
- CreateplaceviewCept – тестування форми вводу інформації з умовами спостережень за птахами на маршруті;
- CreateplaceviewplaceCept - тестування форми вводу інформації з умовами спостережень за птахами на площадці;
- CreatereportwatchCept – тестування форми вводу інформації з результатами обліку на пунктах стопстереження.

Створені сценарії автоматизованих тестів включає в себе певний набір правил або шагів, при створенні тестів для WEB-порталу обліку та моніторингу міграції птахів в Азово-Чорноморському регіоні України.

Тестування сторінок з формами вводу певної інформації (в назві тесту мають слово «Create»), буде складатися із функцій, які будуть вводити в форму не тільки коректні дані, а й з помилками, для того, щоб перевірити коректність виводу попереджень. Розглянемо на прикладі тесту з реєстрації користувачів порталу, де обов'язковими полями є: логін, email та пароль.

Тестування сторінок з текстовою інформацією, або загальний тест сторінок, складається з пошуку елементів, які характерні для вказаної сторінки, так для головної сторінки це посилання на кнопку «Реєстрація» (рис. 1) та «Вхід».

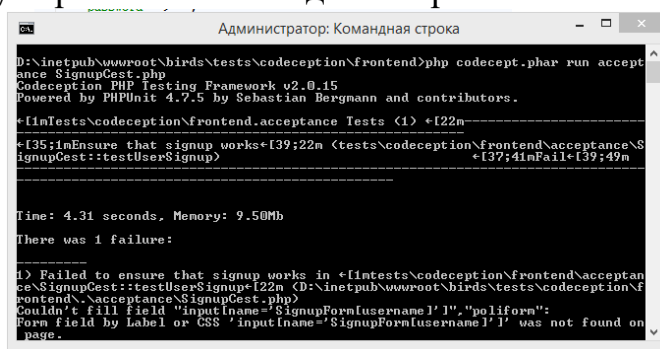
```
$I->wantTo('ensure that signup works');

$signupPage = SignupPage::openBy($I);
$I->amGoingTo('submit signup form with not correct email');
$signupPage->submit([
    'username' => 'poliform',
    'email' => 'testtest.ru',
    'password' => 'protege',
]);
$I->see('Email is not a valid email address.');
```

```
$I->amGoingTo('submit signup form with not correct password');
$signupPage->submit([
    'username' => 'poliform',
    'email' => 'testtest.ru',
    'password' => '',
]);
$I->see('Пароль cannot be blank.');
```

```
$I->wantTo('submit signup form with all correct data');
$signupPage->submit([
    'username' => 'poliform',
    'email' => 'testtest.ru',
    'password' => 'protege',
]);
$I->see('Вы успешно зарегистрированы');
```

Рис. 1. Сценарій перевірки вводу даних форми реєстрації
Результат тесту отримали в командній на рис. 2.



```
Администратор: Командная строка

D:\inetpub\wwwroot\birds\tests\codeception\frontend>php codecept.phar run acceptance SignupCest.php
Codeception PHP Testing Framework v2.0.15
Powered by PHPUnit 4.7.5 by Sebastian Bergmann and contributors.

+-----+
+ [InTests\codeception\frontend\acceptance Tests <1>] +-----+
+ [35:inEnsure that signup works+ [39:22m <tests\codeception\frontend\acceptance\SignupCest::testUserSignup> + [37:41mFail+ [39:49m
+-----+

Time: 4.31 seconds, Memory: 9.50Mb

There was 1 failure:

1) Failed to ensure that signup works in + [InTests\codeception\frontend\acceptance\SignupCest::testUserSignup+ [22m <D:\inetpub\wwwroot\birds\tests\codeception\frontend\acceptance\SignupCest.php>
Could't fill field "inputname="SignupForm[username]"", "poliform":
Form field by Label or CSS "inputname="SignupForm[username]" was not found on page.
```

Рис. 2. Результат перевірки форми реєстрації

Тестування форми авторизації відбувається аналогічно перевірки форми с даними, обов'язковою частиною, є визначення інформації яка виведеться після успішної авторизації.

Висновки. Web-портал обліку міграції птахів в Азово-Чорноморського регіону, був розроблений на платформі Yii Framework 2.x, яка включає в собі програмний пакет з автоматизованими тестами Codeception, який дає змогу проводити тестування різних видів: функціональне, unit-тести та приймальні тести.

Використання різних видів тестування при розробці web-порталу дало можливість визначити та задокументувати основні проблемні ділянки системи, з метою подальшого вдосконалення системи. Слід зауважити, що автоматизовані тести вирішують багато проблемних питань, але вони не дають змогу повно і якісно перевірити на коректність розміщення блоків на сайті та з точки зору юзабіліті сервісу. Такі параметри залишаються на розсуд людини-тестувальнику.

Літератури

1. Абдикеев Н. М. Проектирование информационных систем (современные технологии): Учебное пособие. / Н. М. Абдикеев. – М. : КОС-ИНФ, Рос. экон. акад., 2003. – 140 с.

2. Ахтырченко К.В. Методы и технологии реинжиниринга ИС / К.В. Ахтырченко, Т.П. Сорокваша / Труды Института Системного Программирования РАН. – [Электронный ресурс]. – Режим доступа: <http://citforum.ru/SE/project/isr>.
3. База данных "Онлайн дневники наблюдений" [Электронный ресурс] – Режим доступа: <http://ru-birds.ru/index.php/ru>.
4. Башмаков А.И. Интеллектуальные информационные технологии: Учеб. пособие. / А.И. Башмаков, И.А. Башмаков. – М. : Изд-во: МГТУ им. Н.Э. Баумана, 2005. – 304 с.
5. Васвани В. Разработка веб-приложений на PHP / В. Васвани–Питер, 2012. – 432 с.
6. Kohana, Yii, Symfony, CodeIgniter. – [Электронный ресурс]. – Режим доступа: <http://www.whyrupal.ru/sravnenie-php-freimvorkov-kohana-yii-symfony-codeigniter>
7. Yii-framework. Часть 1. Введение: – [Электронный ресурс] – Режим доступа: <http://www.Linkexchanger.su/2010/418.html>

Анотація: В статті проаналізовані види автоматизованого тестування, які використовувалися при розробці web-порталу обліку міграції птахів. Автором приведена загальна характеристика функціональних, unit-тестів та приймальних тестів, їх можливості та приклади їх використання. Також визначені спектр проблем, які можуть бути визначені за допомогою автоматизованих тестів.

Ключові слова: інформаційна система; моніторинг міграції; автоматизовані тестування; unit-тести; функціональне тестування.

Аннотация. В статье проанализированы основные виды автоматизированного тестирования, которые использовались при разработке web-портала учета миграции птиц. Автором приведена общая характеристика функциональных, unit-тестов и приемных тестов, их возможности и примеры использования. Также определены спектр проблем, которые могут быть решены с помощью автоматизированных тестов.

Ключевые слова: информационная система; мониторинг миграции; автоматизированные тестирования; unit-тесты; функциональное тестирование.

Summary. The article analyzed the types of automated tests that were used in the development of web-portal account migration. The author shows the general characteristics of functional, unit-tests and acceptance tests, their capabilities and examples of their use. Also defined range of problems that can be identified by automated test.

Keywords: Information system; monitoring of migration; automated testing; unit-tests; functional testing.